Hepatoprotective effects of systemic ER activation

BulkRNAseq - Differential expression analysis

Christian Sommerauer & Carlos Gallardo

25 July, 2023

```r
# library import
library(tidyverse)
library(DESeq2)
library(edgeR)
```

Load data

```r
# removed outlier sample PPT_HFD_male_4 for differential expression (see PCA plot, fig. 1C)
# raw counts RNAseq
raw_counts <- read.table(
  file = 'data/bulkRNAseq_mmus_rawcounts.tsv',
  stringsAsFactors = FALSE,
  sep = '\t',
  header = TRUE) %>%
  dplyr::select(-PPT_HFD_male_4) %>%
  tibble::column_to_rownames('geneID') %>%
  as.matrix()

# design RNAseq
design_meta <- read.table(
  file = 'data/bulkRNAseq_mmus_design.tsv',
  stringsAsFactors = FALSE,
  sep = '\t',
  header = TRUE) %>%
  filter(sample != 'PPT_HFD_male_4')

# ensembl gene annotation (Mus musculus)
gene_ann <- read.table(
  file = 'data/ensembl_mmus_sep2019_annotation.tsv',
  stringsAsFactors = FALSE,
  sep = '\t',
  header = TRUE,
  fill = FALSE,
  quote = '')
```

Differential expression

Run DESeq2 pipeline

```r
ds_data <- DESeqDataSetFromMatrix(countData = raw_counts,
                                  colData = design_meta,
                                  design = ~ 0 + condition)
ds_data <- estimateSizeFactors(ds_data)
ds_data <- DESeq(ds_data)

# filtering to minimum 15 counts in at least 8 samples
ds_data <- ds_data[rowSums(counts(ds_data, normalized=TRUE) >= 15 ) >= 8, ]

# annotations for gene background
gene_ids_bg <- rownames(counts(ds_data, normalized=TRUE))
gene_ann_bg <- gene_ann %>%
  filter(ensembl_gene_id %in% gene_ids_bg)

# comparisons
```

```
DESeq2_DEGs <- list(
  CDfVsCDm = results(ds_data, contrast = c('condition', 'CDf', 'CDm')),
  HFDfVsHFDm = results(ds_data, contrast = c('condition', 'HFDf', 'HFDm')),
  CDfVsHFDf = results(ds_data, contrast = c('condition', 'CDf', 'HFDf')),
  CDmVsHFDm = results(ds_data, contrast = c('condition', 'CDm', 'HFDm')),
  DPNVsHFDm = results(ds_data, contrast = c('condition', 'DPN', 'HFDm')),
  DIPVsHFDm = results(ds_data, contrast = c('condition', 'DIP', 'HFDm')),
  E2VsHFDm = results(ds_data, contrast = c('condition','E2', 'HFDm')),
  PPTVsHFDm = results(ds_data, contrast = c('condition','PPT', 'HFDm'))
)

# add annotation to DEG lists
DESeq2_DEGs <- lapply(DESeq2_DEGs, as.data.frame)
DESeq2_DEGs <- lapply(DESeq2_DEGs, rownames_to_column, var = 'ensembl_gene_id')
DESeq2_DEGs <- lapply(DESeq2_DEGs, inner_join, y = gene_ann, by = 'ensembl_gene_id')
```

Run edgeR pipeline

```
groups <- design_meta$condition
dge <- DGEList(raw_counts, group = groups)
design <- model.matrix(~0 + groups)

# filter on CPM
dge <- dge[(rowSums(cpm(dge) > 1) >= 8), ]

y_dge <- calcNormFactors(dge, method = 'TMM')
y_dge <- estimateGLMCommonDisp(y_dge, design)
y_dge <- estimateGLMTrendedDisp(y_dge, design)
y_dge <- estimateGLMTagwiseDisp(y_dge, design)

fit_dge <- glmFit(y_dge, design)

# comparisons
edgeR_DEGs <- list(
  CDfVsCDm = glmLRT(fit_dge, contrast = makeContrasts(groupsCDf-groupsCDm, levels = design)),
  HFDfVsHFDm = glmLRT(fit_dge, contrast = makeContrasts(groupsHFDf-groupsHFDm, levels = design)),
  CDfVsHFDf = glmLRT(fit_dge, contrast = makeContrasts(groupsCDf-groupsHFDf, levels = design)),
  CDmVsHFDm = glmLRT(fit_dge, contrast = makeContrasts(groupsCDm-groupsHFDm, levels = design)),
  DPNVsHFDm = glmLRT(fit_dge, contrast = makeContrasts(groupsDPN-groupsHFDm, levels = design)),
  DIPVsHFDm = glmLRT(fit_dge, contrast = makeContrasts(groupsDIP-groupsHFDm, levels = design)),
  E2VsHFDm = glmLRT(fit_dge, contrast = makeContrasts(groupsE2-groupsHFDm, levels = design)),
  PPTVsHFDm = glmLRT(fit_dge, contrast = makeContrasts(groupsPPT-groupsHFDm, levels = design))
)

# calculate fdr and add annotation to DEG lists
edgeR_DEGs <- lapply(edgeR_DEGs, function(x) as.data.frame(x$table))
edgeR_DEGs <- lapply(edgeR_DEGs, rownames_to_column, var = 'ensembl_gene_id')
edgeR_DEGs <- lapply(edgeR_DEGs, function(x) mutate(x, padj = p.adjust(PValue, method = 'fdr')))
edgeR_DEGs <- lapply(edgeR_DEGs, inner_join, y = gene_ann, by = 'ensembl_gene_id')
```

Common DEGs (DESeq2 & edgeR)

```
# filter DESeq2 results
DESeq2_DEGs_filt <- lapply(DESeq2_DEGs, function(x) filter(x, abs(log2FoldChange) > log2(1.75) & padj < 0.05))

# filter edgeR results
edgeR_DEGs_filt <- lapply(edgeR_DEGs, function(x) filter(x, abs(logFC) > log2(1.75) & padj < 0.05))

# intersect DEGs
common_DEGs_filt <- mapply(function(a, b) filter(a, ensembl_gene_id %in% b$ensembl_gene_id), DESeq2_DEGs_filt, edgeR_DEGs_filt, SIMPLIFY=FALSE)
```

Export DEGs

```
DEGs <- list(unfilt = DESeq2_DEGs,
             filt = common_DEGs_filt)

saveRDS(DEGs, file = 'results/bulkRNAseq_mmus_DEGs.rds')
```

SessionInfo

```
sessionInfo()
```

```
## R version 4.0.5 (2021-03-31)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 10 x64 (build 19045)
## 
## Matrix products: default
## 
## locale:
## [1] LC_COLLATE=English_United States.1252 
## [2] LC_CTYPE=English_United States.1252   
## [3] LC_MONETARY=English_United States.1252
## [4] LC_NUMERIC=C                          
## [5] LC_TIME=English_United States.1252    
## 
## attached base packages:
## [1] parallel  stats4    stats     graphics  grDevices utils     datasets 
## [8] methods   base     
## 
## other attached packages:
##  [1] edgeR_3.32.1                limma_3.44.3               
##  [3] DESeq2_1.30.1               SummarizedExperiment_1.20.0
##  [5] Biobase_2.48.0              MatrixGenerics_1.2.1       
##  [7] matrixStats_0.58.0          GenomicRanges_1.42.0       
##  [9] GenomeInfoDb_1.26.7         IRanges_2.24.1             
## [11] S4Vectors_0.28.1            BiocGenerics_0.36.1        
## [13] forcats_0.5.1               stringr_1.4.0              
## [15] dplyr_1.1.2                 purrr_0.3.4                
## [17] readr_2.1.2                 tidyr_1.2.0                
## [19] tibble_3.2.1                ggplot2_3.3.3              
## [21] tidyverse_1.3.1            
## 
## loaded via a namespace (and not attached):
##  [1] bitops_1.0-6           fs_1.5.2               lubridate_1.8.0       
##  [4] bit64_4.0.5            RColorBrewer_1.1-3     httr_1.4.2            
##  [7] tools_4.0.5            backports_1.4.1        utf8_1.1.4            
## [10] R6_2.5.1               DBI_1.1.3              colorspace_2.0-0      
## [13] withr_2.5.0            tidyselect_1.2.0       bit_4.0.4             
## [16] compiler_4.0.5         cli_3.6.1              rvest_1.0.2           
## [19] xml2_1.3.3             DelayedArray_0.16.3    scales_1.2.1          
## [22] genefilter_1.70.0      digest_0.6.27          rmarkdown_2.14        
## [25] XVector_0.30.0         pkgconfig_2.0.3        htmltools_0.5.2       
## [28] dbplyr_2.1.1           fastmap_1.1.0          rlang_1.1.1           
## [31] readxl_1.4.0           rstudioapi_0.13        RSQLite_2.2.3         
## [34] generics_0.1.3         jsonlite_1.8.0         BiocParallel_1.22.0   
## [37] RCurl_1.98-1.2         magrittr_2.0.3         GenomeInfoDbData_1.2.4
## [40] Matrix_1.4-1           Rcpp_1.0.7             munsell_0.5.0         
## [43] fansi_0.4.2            lifecycle_1.0.3        stringi_1.5.3         
## [46] yaml_2.2.1             zlibbioc_1.34.0        grid_4.0.5            
## [49] blob_1.2.4             crayon_1.5.1           lattice_0.20-41       
## [52] splines_4.0.5          haven_2.5.0            annotate_1.68.0       
## [55] hms_1.0.0              locfit_1.5-9.4         knitr_1.31            
## [58] pillar_1.9.0           geneplotter_1.68.0     reprex_2.0.1          
## [61] XML_3.99-0.5           glue_1.4.2             evaluate_0.21         
## [64] modelr_0.1.8           vctrs_0.6.3            tzdb_0.3.0            
## [67] cellranger_1.1.0       gtable_0.3.3           assertthat_0.2.1      
## [70] cachem_1.0.3           xfun_0.31              xtable_1.8-4          
## [73] broom_0.8.0            survival_3.2-7         AnnotationDbi_1.52.0  
## [76] memoise_2.0.1          ellipsis_0.3.2
```